AMENDMENTS TO THE SPECIFICATION

On pages 2-3, please replace paragraphs [0009]-[0012] as follows:

[0009] With reference to figure 1, there is shown a flow diagram of the normal execution of an application program 101 where the application program 101 links to the library.DLL 103 programs which support the execution of the application program. The application program then begins executing the API code 107. Attackers using various debugging tools can place a break point 105 at the beginning of the API code 107 and stop the API calls and thus the execution of the application program 101. This has serious consequences where the application program 101 concerns the security and safety of the CPU and its data.

[0010] With reference to Figure 2, there is shown a flow diagram of the execution of an application program 201 in the preferred embodiment of the present invention. The present invention avoids the breakpoint 211 by copying up to 16 instructions from the original API function to a local buffer by means of a tracer function. The buffered code or replicated code is executed 209 and then control is passed back to the API function 213 at the 17^{th} instruction. In this way any breakpoint 211 set at the beginning of the API is bypassed.

[0011] During startup of an executable (or DLL) 203, the tracer function takes control before the original entry point is reached 215. During this period it copies 207 up to 16 instructions from each protected API to a local buffer within the context of the executing application.

[0012] The tracer function achieves this by tracing 205 into the API code until it reaches the 16^{th} instruction 207, or until an instruction is reached which it cannot follow. An example of an instruction it cannot follow is shown below:

Example 1:

mov esi.0x00000072

mov edx. [0x12345678]

jmp edx

On pages 3-4, please replace paragraph [0014] as follows:

[0014] The table below shows the difference between normal code execution and replicated code execution 209:

Normal code flow	**Replicated code flow**
Push ebp	Push ebp
Mov ebp, esp	Mov ebp, esp
Push 0	Push 0
Push 0487654h	Push 0487654h
Mov fs:0, esp	Mov fs:0, esp
Push 12345678h	Push 12345678h
Push 1	Push 1
Call myfunction	Push Done
Done:	Push ebp
Ret	Mov ebp, esp
myfunction:	Mov eax, [ebp+8]
Push ebp	Mov ebx, [ebp+c]
Mov ebp, esp	Leave
Mov eax, [ebp+8]	Add esp, 08h
Mov ebx, [ebp+0c]	ret
Leave	
Ret 08h	